POWER MINIMIZATION BY SCANCHAIN REORDERING
A Design Report

Term Project on
VLSI Testing and Verification

BY
GROUP 22

Arnab Sinha
Department of Computer Science and Engineering,
Indian Institute of Technology,
Kharagpur
&
Debrup Das
Department of Electrical Engineering,
Indian Institute of Technology,
Kharagpur

Done under the guidance of

Prof. Indranil Sengupta
Department of Computer Science & Engineering,
Indian Institute of Technology,
Kharagpur

Contents

1. Introduction

The system-on-chip (SoC) revolution in parallel with the rising complexity of VLSI circuits has made the issue of automated testing inevitable. Power dissipation in the testing phase is a major challenge for the testing engineers [1]. Essentially, introduction of scan cells in the circuit, is one of the most widely used and accepted way of testing. Re-ordering of these scan cells significantly reduces the power consumption during the testing. In our project, we strive to minimize the power by the application of a two-phase heuristic procedure which can be exploited by any chip-layout program during the placement and routing of scan cells.

2. Problem Description and Analysis

2.1 Problem Statement

For a given a sequential circuit (in Verilog netlist form) and a set of test vectors, reorder the flip-flops in the scan chain, so that power dissipation becomes minimum. Power can be estimated by computing the number of signal transitions occurring in the various lines of the circuit.

2.2 Terminologies and Analysis

In this context we need to define the metrics of power consumption of a digital circuit. These concerns involve energy, average power, instantaneous power and peak power [2].

- ***Energy***: the total switching activity generated during test application. It has direct impact on the battery lifetime.
- ***Average Power***: the total distribution of power over a time period. High average power adds to the thermal load and it should be vented away from the device under test.
- ***Instantaneous Power***: the value of power consumed at any given instant. Elevated instantaneous power may overload the silicon or package power distribution systems and burn-out phenomenon might take place.
- ***Peak Power***: the highest value of power at any given instant. The peak power determines the thermal and electrical limits of components.

In this project we are more interested in optimizing total power or energy consumption. Generally, power consumption is more in the test-mode than in normal mode due to following reasons [3]. First, the design under test (DUT) has a circuitry embedded to reduce the test-complexity is often idle during the normal operations but used extensively in the testing mode. Second, the test efficiency shows a high correlation with the toggle rate. Third, in a circuit, parallel testing is frequently employed to reduce application time. All these factors contribute in the elevated power dissipation in the circuit. Essentially, the energy consumption of a circuit would be decreased if it is possible to minimize the toggle rate.

2.3 Assumptions and Objective

In our project we assume that the entire *ordered* test-set and the circuit is given. We also assume that there is a single scan chain in the circuit. As there exists a high correlation (experimentally shown in [4]) between the switching activities in the internal nodes of the circuit with the transitions taking place in the scan cells we further assume that the primary inputs are *directly controllable* and all the switching activities in the circuit is due to transitions in the scan cells. Our *objective* is two-fold. They are the following.

(i) To determine the order of interconnection between the scan cells such that the total power consumption due to toggling is minimized.

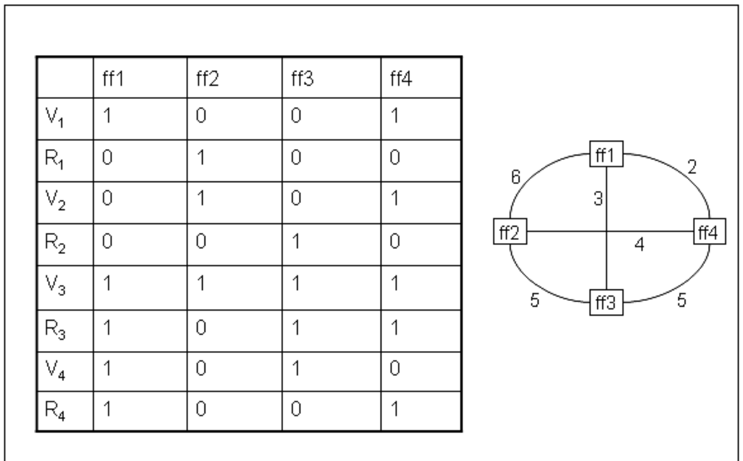
(ii) To identify the input and output scan cells in the scan chain.

3. Methodology

We adopt a two-step heuristic methodology in order to tackle this NP-hard problem. First we shall determine the ordering of the scan cells followed by identification ~~of t~~he scan-in and scan-out cells.

3.1 Determination of chain-order of the scan cells

Consider the circuit with four flip-flops. We consider four test-cases and their corresponding responses. We construct an acyclic graph from the test-sequence and their respective responses. The vertices in the graph represent the flip-flops and the edges represent the transitions between each pair of flip-flops. The edge-weights represent the total bit-difference between them as obtained from the table.



	ff1	ff2	ff3	ff4
V_1	1	0	0	1
R_1	0	1	0	0
V_2	0	1	0	1
R_2	0	0	1	0
V_3	1	1	1	1
R_3	1	0	1	1
V_4	1	0	1	0
R_4	1	0	0	1

Fig 1. Example test sequence and weighted graph

Hence the problem reduces to determination of a Hamiltonian cycle in the constructed acyclic graph. This problem is Traveling Salesman Problem, which is well known to be NP-hard (the number of possible solutions is (n-1)!/2 , n being the scan chain length). Among the number of polynomial-time approximation algorithms [5], we choose the greedy algorithm. The greedy algorithm starts from any scan cell which is ff1 in our case, as the choice of initial state is not so crucial when the number of scan cells in the graph is considerably high [6]. From any given node the algorithm chooses the least weighted edge leading to an undiscovered flip-flop. The final series in this case is **ff1-ff4-ff2-ff3-ff1**. The complexity of the algorithm is **O(n^2)**, as for each node all its incident edges are considered.

3.2 Identification of Scan-in and Scan-out cells

Let us consider the oriented cyclic graph reflecting the obtained order from the previous step of the heuristic, shown in fig 2.At this stage cutting the cyclic graph at each edge will give us the scan-in and scan-out flip-flops. Each cutting solution differs in number of transitions and propagation and the one with least number of transitions will give the optimal solution to our problem.

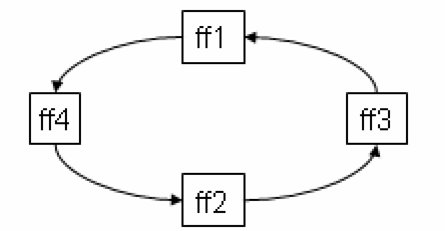


Fig 2. Oriented Cyclic Graph

Here we define a new metric for counting cost of transition and propagation of a given test-vector. Consider the example shown in fig 3. We define the metric of weighted transition as the following,

Weighted Transitions = Σ(Size of scan chain – position of transition)

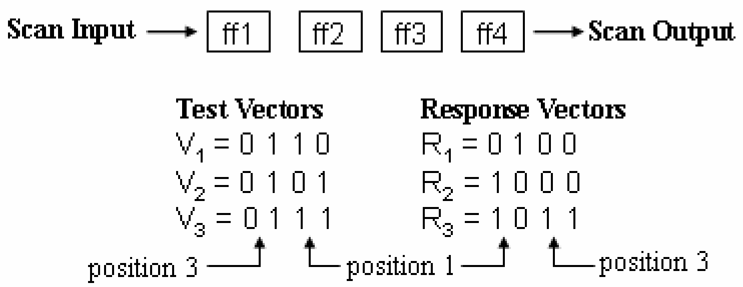


Fig 3. Example set of test vectors and their responses

It was shown in [4] that this metric is good measure for estimating the power dissipation in DUT while testing is done through scan cells. Considering the first test vector, we get the weighted transition for V_1, (4-3) + (4-1) = 4, and that of R_1, (4-1) + (4-2) = 5. Similarly we get 6, 3, 1, 5 for V_2, R_2, V_3, R_3 respectively. Assuming that the initial state was '0000', the total number of weighted transitions produced by the test vectors and their responses,

(4 + 5 + 6 + 3 + 1 + 5) + 4 = 28.

This measure of this metric gives us an estimate of the power consumption due to toggling. So if the value of weighted transition is high, the associated power consumed is also bound to be high.

Following this idea, we compute the weighted transition for each cutting solution. We obtained the following result.

ff1-ff4-ff2-ff3	19+4=23
ff3-ff1-ff4-ff2	21+4=25
ff2-ff3-ff1-ff4	23+4=27
ff4-ff2-ff3-ff1	25+4=29

Hence, the ordering "**ff1-ff4-ff2-ff3**" gives the optimal solution in this case, with ff1 and ff3 being the scan-in and scan-out cells respectively. This heuristic has a complexity of O(n), n being the number of flip-flops in the circuit. So the overall time complexity of the two-step heuristic is $O(n^2)$.

4. Implementation Idea

The following fig 4 is self-illustrative in depicting the data as well as control flow of the project. We start with preprocessing the bench file, followed by the test-pattern generation. Next we extract the number of flops and the binary patterns which is fed to the TSP generator which constructs the graph. The graph is taken as input in the TSP Solver which then outputs the cyclic oriented graph, i.e. an optimal Hamiltonian Cycle. Next we compute the scan-in and scan-out flops followed by the generation of final Verilog file with optimal ordering of the scan-chains.

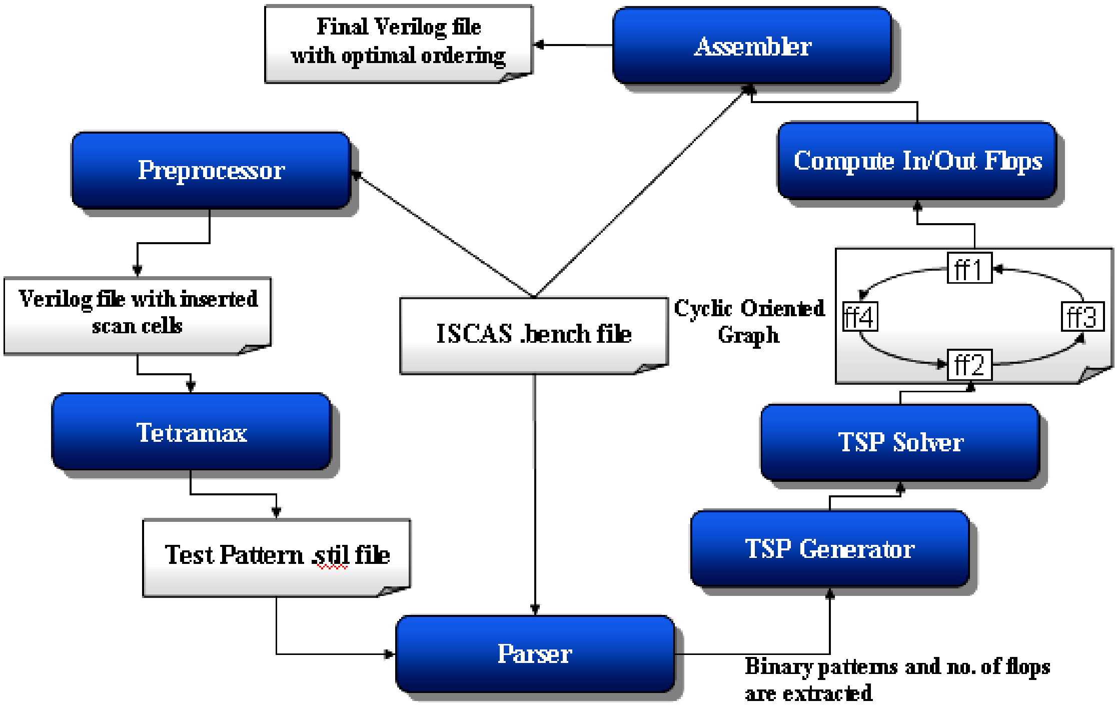
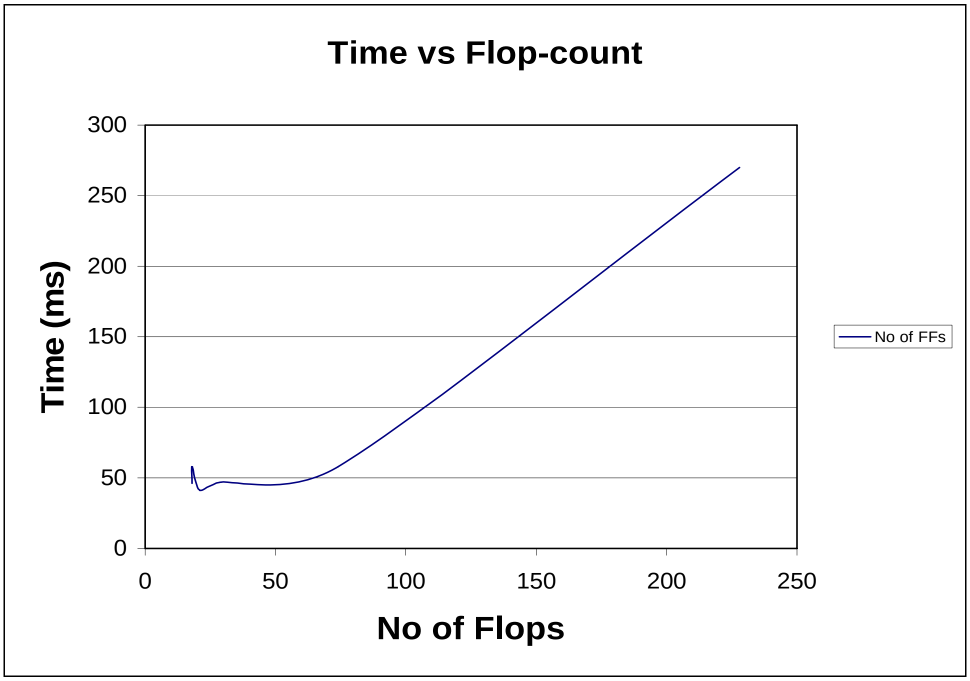


Fig 4. The flow of the CAD tool

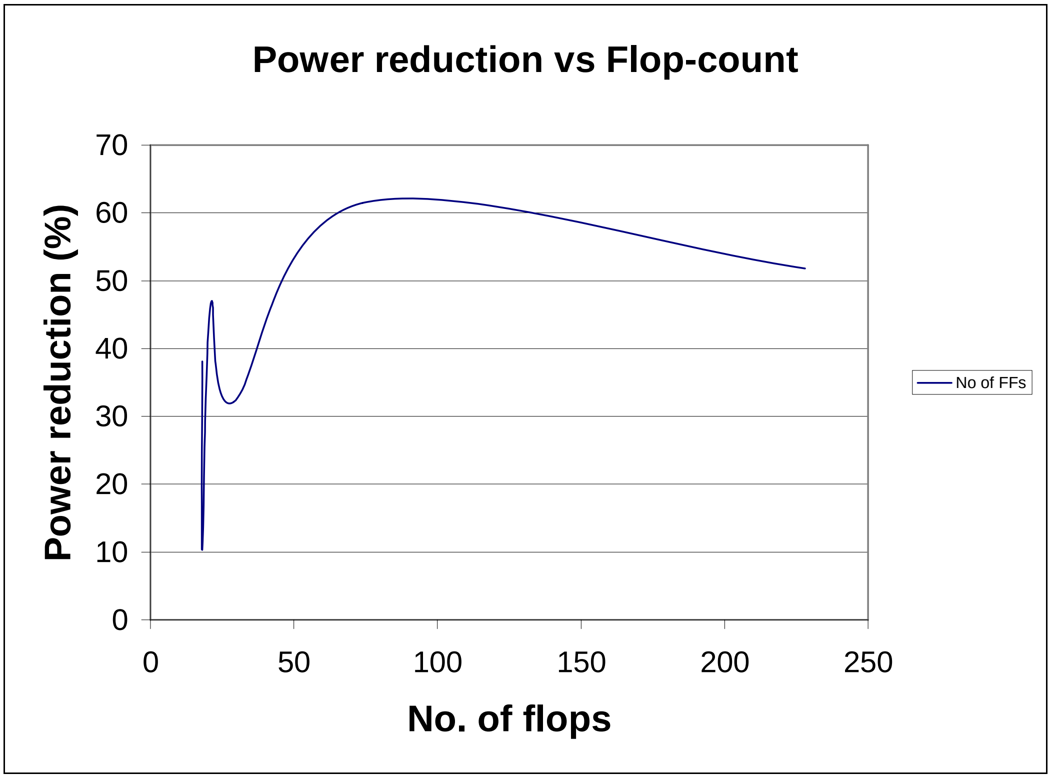
5. Comparative results

In this section we present a comparative result obtained from test-bench circuits of various degree of complexity.

	s1196.bench	s1238.bench	s444.bench	s953.bench	s1423.bench	s9234.bench
Flop Count	18	18	21	29	74	228
Power before Optimization (*toggle count*)	2510	2729	931	6489	14818	300293
Power after Optimization (*toggle count*)	1554	2445	497	4407	5714	144721
Power Reduction *((Row1-Row2)/Row1*100*)	38.08	10.40	46.62	32.09	61.44	51.81
Time Taken (*ms*)	18	58	41	47	58	270



The curve shows that there is a steady rise in the amount of time required with the increasing number of flops in the circuit which is expected.



The graph shows that with the increasing number of flop-counts there is an initial steep rise in the power reduction, followed by a steady fall in the power reduction achieved. This invariably points to the fact that the circuit's inherent complexity (e.g. the number of gates present) should be considered which we have ignored in these results. Nevertheless, these results faithfully represent the situation when we use scan-and-hold scan flops.

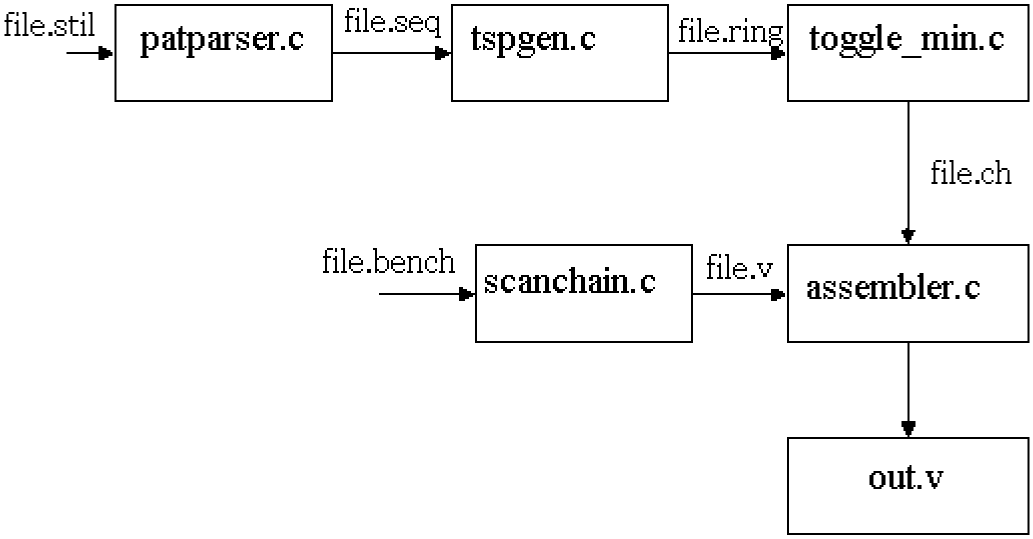
6. Conclusion

Traditionally, researchers have dealt with this issue of power dissipation during testing, in a different light. They have implemented smart ordering of test-vectors [7], [8], [9]. Scan cell ordering has been addressed in [7], where a random ordering heuristic and simulated annealing had been applied. The major advantage in this methodology is, there is no penalty in the circuit performance. The area overhead due to routing area can be kept reasonably low when managed by a efficient layout synthesis program. Also, due to its very low time complexity it is suitable and easy to use during fast design-space verification.

7. Manual

The following C-files are used to generate the desired output:

1. scanchain.c
2. patparser.c
3. tspgen.c
4. hamcycle.c
5. toggle_min.c
6. assembler.c



_

The functions of each of the files are described below:

1. **scanchain.c:** This file takes the .bench as input and gives the corresponding verilog file (.v file) as the output.
2. **patparser.c:** The output of the design analyzer after scan chain insertion is saved in .stil file. This .stil file is fed as input to the program patparser.c which parses the file and extracts information about the test patterns to be applied to the circuit. The output is stored in a .seq file.
3. **tspgen.c:** It takes in the .seq file and writes the output in a .gr file. The program tspgen.c generates a NxN matrix where N=number of scan flip flops in the circuit. The value of the element a[i][j] is given by:
   a[i][j]= -1 if i=j;
   a[i][j]= a[j][i] = total bit-difference between i^{th} and j^{th} flip-flop for all test vectors and their expected responses.

4. **hamcycle.c:** The .gr file is processed by the program hamcycle.c. The program uses the greedy algorithm to find the Hamiltonian cycle of the weighted graph. The output is stored in a .ring file.
5. **toggle_min.c:** It takes the .ring file as the input and gives the .ch file as the output. This program finds the best possible scanchain out of the N possible scanchains that can be generated from the ring of scanchains.
6. **assembler.c:** The assembler.c takes in the .bench, .v and the .ch files and writes the final verilog file with the best possible scanchain inserted.

7.1 How to Compile and Run the Software

For ease a Makefile has been created which runs the above files in a sequential manner. The different modules of the Makefile in the order of their dependence are:

1. clean: removes all intermediate files.
2. Scanchain: runs scanchain.c.
3. TSP_Gen: runs tspgen.c.
4. Ham: runs ham.c.
5. Toggle: runs toggle_min.c.
6. gen: runs assembler.c.

The lower modules are dependent on higher ones. Thus execution of module Scanchain results in execution of modules clean and Scanchain. In order to compile the code,

```
$ make
```

In order to execute the compiled code,

```
$ make –i gen “FILE=s444” “BENCH_FILE = s444.bench” “VERILOG_FILE = s444.v”
```

where, inputs are s444.bench and s444.stil and the output is s444.v in the same directory.

8. References

[1] A. Crouch, "Design-for-Test for Digital IC's and Embedded Core Systems", Prentice Hall ISNB 0-13-084827- 1, 1999.

[2] Y. Bonhomme, P. Girard, C. Landrault, S. Pravossoudovitch, “Power Driven Chaining of Flip-flops in Scan Architecture”

[3] Y. Zorian, "A Distributed BIST Control Scheme for Con~plexV LSI Devices", IEEE VLSI Test Symp., pp. 4-9, 1993.

[4] R. Sankaralingam, R. Oruganti and N. Touba, "Static Compaction Techniques to Control Scan Vector Power Dissipation", IEEE VLSI Test Symp., pp. 35-42,2000.

[5] D.S. Johnson and L.A. McGeoch, "The Traveling Salesman Problem : A Case Study in Local Optimization", in Local Search algorithms in Combinatorial Optimization, E.H.L. Aarts and J.K. Lenstra, eds. John Wiley and Sons, 1996.

[6] M. Gondran and M. Minoux, "Graphes et Algorithmes", Editions Eyrolles, ISSN 0399-4198, 1979.

[7] V. Dabholkar, S. Chakravarty, I. Pomeranz and S.M. Reddy, 'Techniques for Reducing Power Dissipation During Test Application in Full Scan Circuits", IEEE Transactions on CAD, Vol. 17, **No** 12, pp. 1325-1333, December 1998.

[8] P. Girard, C. Landrault, S. Pravossoudovitch and D. Severac, "Reducing Power Consumption during Test Apphcation by Test Vector Ordering ", IEEE Int. Symp. on Circuits and Systems, CD-Rom proceedings, 1998.

[9] P. Girard, L. Guiller, C. Landrault and S. Pravossoudovitch, "A Test Vector Ordering Technique for Switching Activity Reduction during Test Operation ", IEEE Great Lakes Symp. on VLSI, pp. 24-27, 1999.